

Data compression algorithms for flow tables in Network Processor (RuNPU).

Nikita Nikiforov

Lomonosov Moscow State University

Moscow, Russia

nickiforov.nik@gmail.com

Dmitry Volkanov

Lomonosov Moscow State University

Moscow, Russia

volkanov@asvk.cs.msu.ru

Abstract—This paper addresses the problem of packet classification within a network processor (NP) architecture without the separate associative device. By the classification, we mean the process of identifying a packet by the header. The classification stage requires the implementation of data structures to store the flow tables. In our work we consider NP without the associative memory. In considering NP flow tables represent like an assembly language program. For translating flow tables into assembly language programs, tables translator was used. Nowadays flow tables can take tens of megabytes of memory. This is the reason for implement data compression algorithms in flow table translator. In this work we provide the data compression algorithms: Optimal rule caching, recursive end-point cutting and bit string compression. An evaluation of the implemented data compression algorithms was performed on a simulation model of the NP.

Index Terms—Network processor, software-defined networks, packet classification, data compression.

I. INTRODUCTION

At present, software-defined networks (SDN) are in active developing and require high-performance switches. The main functional element of the high-performance SDN switch is programmable network processor. The network processor is system-on-chip specialized for network packet processing. In our work we consider the programmable NP. By programmable we mean such NP, that supports to change the packet processing program and the set of processed header fields on the fly.

This article will discuss about data compression algorithms used for flow tables. Flow tables needs for packet classification process. Flow table is the set of flows, that defines by OpenFlow protocol. The classification is the process of identification a network packet by it's header.

II. NETWORK PROCESSOR ARCHITECTURE

III. THE PROBLEM

IV. RELATED WORK

V. OUR SOLUTION

VI. EVALUATION

VII. FUTURE WORK